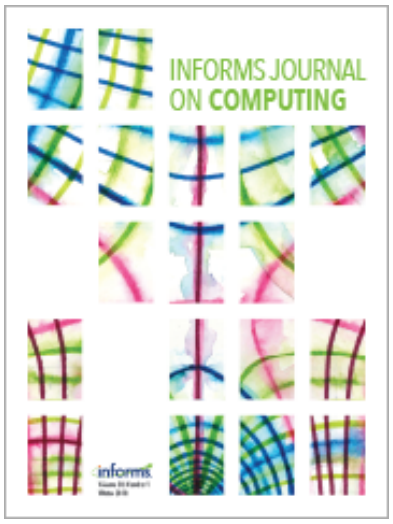


INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Recourse in Kidney Exchange Programs

Bart Smeulders, Valentin Bartier, Yves Crama, Frits C. R. Spieksma

To cite this article:
Bart Smeulders, Valentin Bartier, Yves Crama, Frits C. R. Spieksma (2022) Recourse in Kidney Exchange Programs. INFORMS Journal on Computing 34(2):1191-1206. https://doi.org/10.1287/ijoc.2021.1099

http://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Vol. 34, No. 2, March–April 2022, pp. 1191–1206
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

Recourse in Kidney Exchange Programs

Bart Smeulders,[a] Valentin Bartier,[b] Yves Crama,[c] Frits C. R. Spieksma[a]

[a] Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven 5612 AE, Netherlands; [b] G-SCOP, Grenoble INP Université Grenoble Alpes, Grenoble 38031, France; [c] HEC Management School, University of Liège, Liege 4000, Belgium

Contact: b.smeulders@tue.nl, https://orcid.org/0000-0002-6092-2037 (BS); valentin.bartier@grenoble-inp.fr (VB); yves.crama@uliege.be, https://orcid.org/0000-0002-7470-4743 (YC); f.c.r.spieksma@tue.nl, https://orcid.org/0000-0002-2547-3782 (FCRS)

Received: August 2, 2019
Revised: May 4, 2020; January 28, 2021; April 26, 2021
Accepted: May 7, 2021
Published Online in Articles in Advance: October 28, 2021

https://doi.org/10.1287/ijoc.2021.1099

Abstract. We introduce the problem of selecting patient-donor pairs in a kidney exchange program to undergo a *crossmatch* test, and we model this *selection problem* as a two-stage stochastic integer programming problem. The optimal solutions of this new formulation yield a larger expected number of realized transplants than previous approaches based on internal recourse or subset recourse. We settle the computational complexity of the selection problem by showing that it remains NP-hard even for maximum cycle length equal to two. Furthermore, we investigate to what extent different algorithmic approaches, including one based on Benders decomposition, are able to solve instances of the model. We empirically investigate the computational efficiency of this approach by solving randomly generated instances and study the corresponding running times as a function of maximum cycle length, and of the presence of nondirected donors.

Summary of Contribution: This paper deals with an important and very complex issue linked to the optimization of transplant matchings in kidney exchange programs, namely, the inherent uncertainty in the assessment of compatibility between donors and recipients of transplants. Although this issue has previously received some attention in the optimization literature, most attempts to date have focused on applying recourse to solutions selected within restricted spaces. The present paper explicitly formulates the maximization of the expected number of transplants as a two-stage stochastic integer programming problem. The formulation turns out to be computationally difficulty, both from a theoretical and from a numerical perspective. Different algorithmic approaches are proposed and tested experimentally for its solution. The quality of the kidney exchanges produced by these algorithms compares favorably with that of earlier models.

History: Accepted by J. Paul Brooks, Area Editor for Applications in Biology, Medicine, & Healthcare.
Funding: The research of F.C.R. Spieksma is supported by Dutch Research Council (NWO) Gravitation Project NETWORKS [Grant 024.002.003]. Computational resources were provided by the Consortium des Équipements de Calcul Intensif, funded by Fonds de la Recherche Scientifique–FNRS [Grant 2.5020.11] and the Walloon Region of Belgium.
Supplemental Material: The online appendix is available at https://doi.org/10.1287/ijoc.2021.1099.

Keywords: Benders decomposition • stochastic programming • kidney exchange • healthcare

1. Introduction: Kidney Exchange Programs

Mathematical optimization techniques have established themselves as an important and indispensable tool in guiding decisions in kidney exchange programs. There is a large and fast-growing amount of literature documenting various successful implementations of algorithms that find cycles and chains in appropriately defined *compatibility* graphs. The increased performance of such algorithms has led to a better use of available human kidneys, and as a result, many lives have been positively impacted.

This paper focuses on the issue of dealing with incompatibilities that may reveal themselves *after* an intended transplant has been identified. This is an important issue; for example, Dickerson et al. (2019) report that 93% of proposed matches fail in the United Network for Organ Sharing program, for a wide variety of reasons. In the National Health Service (NHS) Living Kidney Sharing Scheme, 30% of identified matches did not proceed to transplant between 2013 and 2017 (NHS 2017a). Here, we analyze how this phenomenon can be taken into consideration in optimization models. We propose a new, generic, integer programming formulation to identify the maximum expected number of transplants, and we perform extensive computational experiments with this model. The resulting outcomes give insights on how kidney exchange programs can best prepare for the challenges that arise when confronted with a posteriori incompatibilities.

In order to set the stage for our contribution, we first give a stylized description of the operation of a kidney exchange program; in this description, we momentarily ignore many of the practical features that exist in real-life kidney exchange programs.

The preferred treatment for a patient with end-stage renal disease is receiving a kidney from a living human donor. Many patients have a donor, often a friend or family member, who has volunteered to donate one of their kidneys for a transplant. However, a donor must be *compatible* with a patient for a transplant to be possible. Determining whether a donor is compatible with their corresponding patient is done by a preliminary screening, based on blood type and immunological properties. When the donor and the patient are not compatible, they may together decide to enter a kidney exchange program where more transplant opportunities become available by relying on the diversity of a larger pool of individuals. We refer to Gentry et al. (2011) for more information on this process. Thus, a kidney exchange program consists of a set of patient-donor pairs (the *pool*), for which compatibilities have been derived from the preliminary screening tests. The objective is then to identify sequences of potential kidney donations among pairs of the pool, whereby each patient in the sequence receives a kidney from the donor of the previous pair, whereas the associated donor donates a kidney to the next patient. This description implies that the sequence of donations must necessarily be cyclic. Exchange programs, however, may also allow for the presence of *nondirected donors* who are not associated with any patient. They may increase the range of feasible sequences by acting as starting points of sequences of donations that end with a patient receiving a kidney, whereas the associated donor is not involved in any transplant (and may subsequently act as a nondirected donor).

A natural and well-established way of describing the operation of a kidney exchange program is by considering a so-called *compatibility graph*. In this directed graph, a vertex is associated with each patient-donor pair and each nondirected donor. There is an arc from vertex i to vertex j if the donor associated with vertex i is compatible with the patient of pair j (according to preliminary screening). A k-cycle, that is, a directed cycle of length k in this graph, indicates a sequence of k transplants that can be simultaneously performed, based on the compatibilities identified in the preliminary screening phase. Similarly, a k-chain, that is, a directed chain of length k originating at a vertex associated with a nondirected donor, also indicates a feasible sequence of k transplants. Typically, for logistical reasons, an upper bound is given on the length of the cycles and chains that can be considered for transplants. The problem faced by the kidney exchange program is then to find a collection of vertex-disjoint cycles and chains of bounded length, covering as many arcs as possible, thus allowing for the largest possible number of transplants in the current pool (see Section 2.2 for a more formal definition). We will refer to this optimization problem as the *kidney exchange problem* (KEP).

Clearly, the previous description is a very stylized sketch of the operation of a kidney exchange program. In practice, a program may present many other features; for example, arcs may be weighted to prioritize certain types of patients or transplants. We refer to Gentry et al. (2011) and Anderson et al. (2015) for a more elaborate discussion and to Biró et al. (2019) for an overview of kidney exchange programs in Europe.

The key issue that we address here is that, in kidney exchange programs, compatibilities arising from preliminary screening are rarely certain and must be assessed by further tests. Indeed, after solving the kidney exchange problem, that is, after having identified a set of cycles and chains intended to give rise to a number of transplants, it may turn out that, for various reasons, some of the transplants cannot take place. This may be the case because a patient has already received a kidney from another program or is too ill to undergo surgery. Another frequent reason is that further compatibility tests, called *crossmatch tests*, may reveal previously undetected incompatibilities. Such crossmatch tests must always be performed *before* any transplant is performed. They consist of analyzing blood samples of recipient and donor. There is a significant probability that a crossmatch test detects an incompatible transplant (Dickerson et al. 2016). Of course, when this happens, it implies that not only this particular transplant cannot be carried out, but also the other transplants in the same cycle, or further in the chain, fail to be implemented.

Because crossmatch tests are time consuming and expensive, most kidney exchange programs only perform such tests once an intended transplant has been identified (Biró et al. 2019), so that uncertainty remains until that time. In line with previous literature (Glorie 2012, 2014; Alvelos et al. 2015; Klimentova et al. 2016; Dickerson et al. 2019), we model this uncertainty by specifying *probabilities* for the compatibility between a donor and a patient (or for the availability of a patient-donor pair).

Our main contribution in this paper is to explicitly identify the problem of selecting the set of arcs that should undergo the crossmatch tests to maximize the expected number of transplants (which, as mentioned by Glorie 2014, "may be an opportunity to significantly increase the success rates" of kidney exchange programs). We formulate this problem as a two-stage stochastic optimization problem. In the first stage, we select a set of arcs that each will undergo a crossmatch test. The problem of identifying this set of arcs is called the *selection problem*. In the second stage, we

solve the (deterministic) KEP on the graph induced by the arcs that passed the crossmatch tests.

We summarize our contributions as follows.

- We formalize the selection problem and argue that solving this problem is a key ingredient in obtaining solutions that maximize the expected number of transplants (Section 3).
- We establish the computational complexity of the selection problem by proving that it remains NP-hard even when the maximum allowed cycle length (K) is equal to two; this result contrasts with the polynomial solvability of the KEP when $K = 2$. The corresponding theorems are formulated in Section 3, and the proofs can be found in Online Appendix A.
- We show how to apply Benders decomposition to a generic integer programming formulation of the selection problem (Section 4).
- We perform extensive computational experiments illustrating the quality of the solutions found by the selection model compared with other approaches (Section 6) and showing the impact of various modeling and algorithmic choices (Sections 7 and 8).

2. Problem Statement

We provide a brief overview of earlier work on KEPs in Section 2.1. Next, we set the stage by giving a generic formulation of the KEP in Section 2.2 and by detailing the stochastic version of the KEP in Section 2.3.

2.1. Literature Review

Seminal work on modeling kidney exchange programs through integer programming was initiated by Roth et al. (2004, 2006) and Montgomery et al. (2006). The deterministic KEP is a difficult combinatorial optimization problem: as observed in Abraham et al. (2007), it is NP-hard when restricted to any fixed cycle length greater than two. (For cycle length equal to two, KEP can be solved in polynomial time as a maximum matching problem.) In practice, however, optimal solutions of appropriate integer programming formulations can be computed in acceptable running times for medium to large instances; see Dickerson et al. (2016), Mak-Hau (2017), and Manlove and O'malley (2015) for recent references.

To take crossmatch tests and uncertainty into account, two broad classes of approaches have been proposed: *adaptive* and *nonadaptive* approaches. In nonadaptive approaches, a subset of potential transplants is first selected, and crossmatch tests are subsequently performed in parallel on all the arcs of these subsets. The arcs that pass the crossmatch test can finally be used to identify the transplants to be executed by the kidney exchange. Adaptive approaches allow for more rounds of tests. After each round of crossmatch tests, additional arcs are selected for testing and this choice is dependent on the successes and failures in the previous rounds. Eventually, the crossmatching phase terminates and the successful arcs are used to identify the transplants to be performed.

Nonadaptive approaches have received most of the attention in the literature and have led to various formulations of the stochastic problem. Dickerson et al. (2013, 2019) propose reweighting cycles and chains to reflect the expected number of transplants that can be performed using those subgraphs, and they solve the associated weighted packing problem. A compact formulation of this packing problem is given by Dickerson et al. (2016). Pedroso (2014) also reweights cycles, but his model accounts for all arcs induced by each cycle, thus allowing for limited recourse at the cost of additional tests; Alvelos et al. (2015) provide a compact formulation of this packing model. Klimentova et al. (2016) propose another recourse scheme where overlapping cycles can be tested. We give a more explicit description of these recourse schemes in Section 2.3.

Another stream of literature focuses on approximation algorithms. Blum et al. (2013) develop such an algorithm for the case of two cycles where each patient is involved in at most two crossmatch tests; they prove that their algorithm finds near-optimal solutions in sufficiently large kidney exchange pools. Blum et al. (2015) provide nonadaptive approximation algorithms requiring a constant number of tests per vertex and delivering a solution with expected value within a factor $((2/K)^2/(2/K+1))(1-\epsilon)$ of the expected optimal value for the case of K-cycles or K-chains. Assadi et al. (2016), Behnezhad et al. (2019), and other authors subsequently strengthened the result for two cycles.

Adaptive approximation algorithms for bounded cycles and chains have been introduced by Blum et al. (2015). For cycles of length 2, the authors proposed an algorithm that returns a matching with expected value at most $(1-\epsilon)$ the expected optimal matching value after a constant number of rounds and a constant number of queries (i.e., crossmatch tests) per vertex. They further extended these results to the K-set packing problem for which they obtained a $\frac{2}{K}(1-\epsilon)$-approximation algorithm. Assadi et al. (2016) subsequently improved the results for two cycles.

In real-world practice, both adaptive and nonadaptive policies are actually used. The NHS in the United Kingdom computes solutions that can be adjusted if planned transplants do not pass the crossmatch test, in the spirit of the nonadaptive procedure of Pedroso (2014). Specifically, the NHS prefers (ceteris paribus) three cycles with embedded two cycles over three cycles without embedded two cycles. In this way, if one of the transplants in the three cycles turns out to be infeasible, the two cycles can be performed instead

(NHS 2017b). Smaller programs, such as the Dutch and Czech program, iterate between solving a KEP and crossmatching all transplants in the solution. This adaptive process terminates when all tests are successful, and the number of transplants is thus maximized (Biró et al. 2019).

2.2. Deterministic KEP

Let us now turn to a more formal definition of the deterministic KEP. An instance of the problem is defined by a simple, directed graph $G = (V, A)$, and by two integers K and L. Each vertex in V represents either a patient-donor pair or a nondirected donor. An arc $(i, j) \in A$ represents a possible transplant of a kidney from the donor associated with vertex i to the patient associated with vertex j. We use $V(G')$ ($A(G')$) to denote the set of all vertices (arcs) in a subgraph G'. Let C be the set of all directed cycles c in G with length $w_c = |V(c)| = |A(c)|$ at most K. Similarly, let H be the set of all chains h starting from a nondirected donor and with length $w_h = |A(h)| \leq L$.

The KEP is the problem of finding a set of vertex-disjoint cycles of C and chains of H, which maximizes the total number of arcs covered by the set.

Several mathematical programming formulations have been proposed for the KEP (Mak-Hau 2017). For the sake of generality, we are going to assume, here and in the following sections, that KEP is formulated as a 0-1 linear programming problem of the form

$$z(G) = \max \sum_{\ell=1}^{N} a_\ell x_\ell, \tag{1}$$

$$\text{subject to } \mathbf{x} \in P(G), \tag{2}$$

$$\mathbf{x} \in \{0,1\}^N, \tag{3}$$

where $\mathbf{x}$ is a vector of binary variables of appropriate length N that expresses what arcs, cycles, and chains are used for transplants (the exact interpretation depends on the chosen formulation). The vector of coefficients $a \in \mathbb{R}^N$ reflects the lengths of the cycles and chains, and $P(G) \subseteq \mathbb{R}^N$ is a feasible region defined by a finite list of linear inequalities. (In order to simplify the notations, we do not explicitly state the dependence of $z(G)$ and $P(G)$ on K and L.) The generic formulation (1)–(3) emphasizes the fact that any formulation of KEP could be used for our purpose and avoids distraction by the intricacies of such specific formulations. In our computational experiments, we relied on a well-known explicit formulation of KEP, namely, the *position-indexed edge* (PIE) formulation proposed by Dickerson et al. (2016). This formulation is described in Online Appendix B.

2.3. Stochasticity in the KEP

As explained in Section 1, a solution of (1)–(3) may turn out to be unimplementable. Indeed, a potential transplant that has passed preliminary compatibility tests and is part of a selected cycle or chain may not pass the crossmatch test. In order to model this phenomenon, it is customary to introduce, for each arc (i, j), a probability $p_{i,j}$ that the arc passes the crossmatch test and that the intended transplant can proceed (Glorie 2012, 2014; Alvelos et al. 2015; Klimentova et al. 2016; Dickerson et al. 2019). Passing or not passing a crossmatch test may depend, among other factors, on physiological properties of the patient and the donor; historical data can be used to provide reliable estimates of the corresponding probability (Glorie 2012). The events associated with all arcs are assumed to be mutually independent. (A variation of this model occurs when probabilities are associated with vertices, rather than arcs. There is no fundamental distinction and in our computational experiments, we will use vertex probabilities. More generally, one could also imagine a situation where probabilities are specified for subsets of arcs to pass the corresponding crossmatch tests.)

When stochasticity is introduced, it is necessary to specify how the results of the crossmatch tests are used to identify the transplants to be implemented by the exchange program. All nonadaptive strategies share the following generic framework:

- **(Selection)** A subset of arcs, say $T \subseteq A$, is selected for testing.
- **(Testing)** The arcs in T are crossmatched. Let us call R the set of arcs in T that pass the crossmatch test, $R \subseteq T$.
- **(Recourse)** The kidney exchange problem is solved to optimality on the subgraph $G_R = (V, R)$.

In this framework, only the first and third steps are algorithmic: testing is performed by the medical teams and can be viewed as revealing the value of the random Bernoulli variables associated with the arcs in T. (In Blum et al. (2015) or Assadi et al. (2016), this testing step is replaced by "query" instructions.) The third step can be viewed as providing a *recourse* against the outcome of the testing step, and the selection step is usually implemented to maximize the expected value of the solution provided by the recourse, under various restrictions aimed at simplifying the problem. We now describe three different ways of implementing recourse that have been used in literature.

To start with, in Dickerson et al. (2013, 2019), the set T is restricted to consist of a collection of pairwise disjoint cycles and chains. In that case, the model (1)–(3) can be used with a suitable redefinition of the objective function coefficients to express the expected number of transplants in each cycle or chain. In this model, the recourse stage is essentially vacuous, because only those cycles that remain after the crossmatch test, and fragments of chains up to the point of failure, will be implemented. In other words, the model assumes *no*

recourse. However, as convincingly demonstrated in Pedroso (2014) and Klimentova et al. (2016), solutions from the no-recourse model are overly conservative and do not adequately represent the number of transplants that might actually be performed when allowing a limited number of additional crossmatch tests (see Example 1).

Example 1. Consider two subgraphs of a given graph G, say $G' = (V', A')$ and $G'' = (V'', A'')$. We say that G' is *embedded* in G'' if $V' \subseteq V''$. Figure 1 displays an example of a two cycle $(1-3-1)$ embedded in a three cycle $(1-2-3-1)$. Assume that the three cycle is identified by the exchange program in the selection stage, but that arc (1, 2) subsequently fails the crossmatch test, whereas arc (3, 1) passes the test. For an exchange program that allows recourse, it would be possible to further test arc (1, 3), with the hope to be able to implement the two transplants (1, 3) and (3, 1).

The previous observation led Pedroso (2014) and Klimentova et al. (2016) to propose procedures whereby certain types of subgraphs are selected in the first stage, and all arcs embedded in these subgraphs are subsequently crossmatched to provide the input for the recourse stage. As mentioned earlier, in the United Kingdom, the NHS has included limited recourse of this nature (NHS 2017b, Biró et al. 2019).

More specifically, Pedroso (2014) proposes to carry out the selection step by solving the deterministic KEP model (1)–(3) where the binary variables x_c are associated with feasible cycles with appropriately defined cycle weights $a_c = w'_c$, and next to apply crossmatch tests to all arcs embedded in the selected cycles. In case one of the arcs in a selected cycle fails the crossmatch test, recourse can be obtained by selecting an embedded, or internal, cycle instead. This approach is called *internal recourse*. For the model to be correct, the weight w'_c is set equal to the expected optimal value of KEP over the subgraph induced by cycle c. For small enough cycle lengths, these weights can be efficiently computed (Pedroso 2014).

Klimentova et al. (2016) extend the previous idea by solving a packing model similar to (1)–(3), but with variables that are associated with subsets of vertices (instead of cycles) of small size; this approach is accordingly called *subset recourse*. More formally, the integer programming model used in subset recourse can be stated as follows. Let Ω denote the set of all *relevant* subsets of vertices (for a precise definition of relevant subsets, we refer to Klimentova et al. 2016). Define the variables $y_U = 1$ if the vertex subset $U \in \Omega$ is selected for testing, and $y_U = 0$ otherwise. The parameter $a_U = w_U$ denotes now the expected number of transplants that can be realized in the subgraph induced by subset $U \in \Omega$. Then, the selection step of the subset recourse model is formulated as

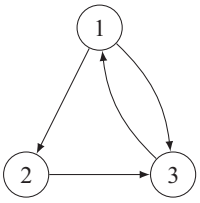
$$\max \sum_{U \in \Omega} w_U y_U, \tag{4}$$

$$\text{subject to} \sum_{U: v \in U} y_U \leq 1 \qquad \forall\, v \in V, \tag{5}$$

$$y_U \in \{0,1\} \qquad \forall\, U \in \Omega. \tag{6}$$

Klimentova et al. (2016) describe methods that allow them to compute the value of w_U when $|U|$ is not too large.

Figure 1. Three Cycle with an Embedded Two Cycle



3. Selection Problem

As shown in the previous section, no-recourse, internal recourse, and subset recourse models only differ in the limitations that they place on the subsets of vertices that are considered in the selection step, and accordingly, on the coefficients used in the KEP model (1)–(3). The choice of these limitations, however, is rather arbitrary and may needlessly restrict the effectiveness of the procedures.

Our proposal in this paper is to focus instead on the key question, which is in our view: what subset of arcs should be selected and subsequently tested for crossmatch? We intend to answer this question by not restricting ourselves to selecting either disjoint cycles and chains, or small disjoint subsets of vertices in the first step. Instead, we propose a model featuring almost no a priori restrictions on the selection step. Indeed, the only condition we impose is an upper bound on the number of crossmatch tests that can be performed, and hence, on the number of arcs that can be selected. This makes sense, because performing crossmatch tests on all possible transplants within the pool of a kidney exchange program, that is, testing all arcs of A, is logistically infeasible.

This leads us to the definition of the following problem, called the *selection problem*. Given a directed graph $G = (V, A)$, we denote by $\mathcal{S} = \{A_1, \ldots, A_m\}$ the collection of all subsets of arcs, arbitrarily numbered, with $m = 2^{|A|}$. We interpret each subset $A_s \in \mathcal{S}$ as a possible *scenario*, that is, as the set of arcs that pass the crossmatch tests under some possible realization of the random variables. (In the sequel, we will often only use the index s to denote a scenario A_s, and we will write $s \in \mathcal{S}$ instead of $A_s \in \mathcal{S}$.) Thus, for each s, the set $A \setminus A_s$ is the set of arcs that would fail the crossmatch tests. The probability q_s of occurrence of

scenario $A_s \in \mathcal{S}$ is computed as

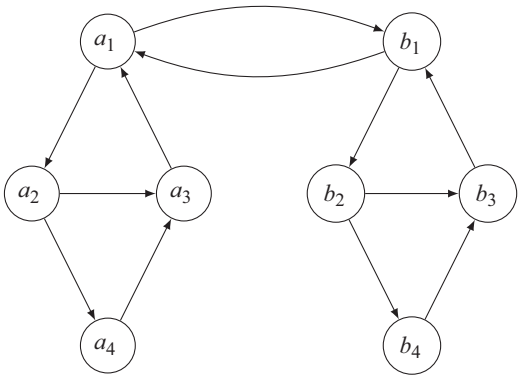
$$q_s := \Big(\prod_{(i,j)\in A_s} p_{i,j}\Big)\Big(\prod_{(i,j)\notin A_s} (1-p_{i,j})\Big).$$

Finally, as mentioned previously, we assume that we are given an upper bound B on the number of crossmatch tests we are allowed to perform. The selection problem is now to identify a subset of arcs $T \subseteq A$, with $|T| \le B$, which maximizes the expected number of transplants in the graph (V, T).

The selection model offers more flexibility than the previous (no-)recourse models, in the sense that every feasible solution of those models yields a feasible solution of the selection problem (provided that the number of arcs to be tested does not exceed B), but not conversely. In fact, for a given budget on the number of arcs to be tested, it is the most flexible model. The difference between an optimal value of the selection problem, and an optimal value of another recourse model indicates how much is lost by restricting the set of feasible solutions (see Example 2 for an illustration).

Example 2. Figure 2 depicts an instance where the presence of restrictions that are inherent to recourse schemes has a negative impact on the expected number of transplants. Assume that the maximum cycle length is $K = 4$, that no chains are allowed, and that the success probability for each arc (i, j) is $p_{i,j} = p = 0.5$. Assume further that we use the subset recourse procedure where Ω contains all subsets of size at most four. Then, the optimal solution of (4)–(6) consists of the subsets $T_1 = \{a_1, a_2, a_3, a_4\}$ and $T_2 = \{b_1, b_2, b_3, b_4\}$. The testing step requires to crossmatch all 10 arcs induced by T_1 and T_2, which leads to an expected 1.0625 transplants. On the other hand, the optimal solution of the selection problem (say, with $B = 10$) would pick instead the eight arcs $(a_1, a_2), (a_2, a_3), (a_3, a_1), (b_1, b_2), (b_2, b_3), (b_3, b_1), (a_1, b_1), (b_1, a_1)$ and would yield an expected 1.1328125 transplants.

Figure 2. Instance of the Selection Problem with $K = 4, p = 0.5$



Our formulation of the selection problem is in the same spirit as the models proposed in Blum et al. (2013, 2015) and Assadi et al. (2016). The main difference is that these authors use local constraints on the number of arcs tested and mostly restrict themselves to two cycles.

In the remainder of this section, we state the computational complexity of the selection problem in Section 3.1, we provide an integer programming formulation in Section 3.2, and we show in Section 3.3 how the sample average approximation technique allows us to arrive at a tractable model.

3.1. Complexity of the Selection Problem

We consider three versions of the selection problem: (i) the selection problem with edge probabilities (called SPedge, see Section 3.1.1), (ii) the selection problem with vertex probabilities (called SPvertex, see Section 3.1.2), and (iii) the selection problem with explicit scenarios (called SPscen, see Section 3.1.3). We state that each of these versions is NP-complete and refer to Online Appendix A for the corresponding proofs.

We point out that a result in Blum et al. (2013) establishes the complexity of a related problem: when given edge probabilities and a bound on the number of edges allowed to be incident to each node, they show the resulting problem to be NP-hard. Our hardness results apply to the restricted case of the selection problem where the only allowable cycles are two cycles, that is, $K = 2$, and where we do not allow chains, that is, $L = 0$. (This is in contrast with the deterministic KEP that can be solved in polynomial time when $K = 2$.) This means that, in this section, we can restrict our attention to compatibility graphs with the property that if arc $(x, y) \in A$, then also $(y, x) \in A$. We model this property by viewing the compatibility graph as an undirected graph consisting of edges instead of arcs. In this view, each edge stands for a two cycle of the original directed graph, and a feasible solution of KEP is a matching.

3.1.1. Selection Problem with Edge Probabilities. We first consider the case where each individual arc in the compatibility graph may, or may not, pass the crossmatch test. Accordingly, in the equivalent undirected model, we assume that each undirected edge is associated with the probability that both corresponding directed arcs pass the crossmatch test. Given a random graph $G = (V, E)$, where each edge $e \in E$ has a success probability $p(e)$ to be present (i.e., to pass the crossmatch tests), we denote by $\mathbb{E}(G = (V, E), p)$ the expected size of a maximum matching on $G = (V, E)$.

We now explicitly state the decision version of this selection problem.

Problem: The decision version of SPedge: DecSPedge.

Instance: A simple, undirected graph $G = (V, E)$, a success probability $p(e)$ $(0 \le p(e) \le 1)$ for each $e \in E$, and numbers B and Z.

Question: Does there exist an edge set $E^* \subseteq E$ such that $|E^*| \le B$ and $\mathbb{E}((V, E^*), p) \ge Z$?

Theorem 1. *DecSPedge is NP-complete.*

3.1.2. Selection Problem with Vertex Probabilities. Assume now that each patient-donor pair (i.e., each vertex) in the compatibility graph has a given probability to be able to participate in an exchange. Then, the KEP with *vertex probabilities* becomes relevant. Given a random undirected graph $G = (V, E)$, where each vertex $v \in V$ has a probability $p(v)$ to be present, we denote by $\mathbb{E}(G = (V, E), p)$ the expected value of a maximum matching on $G = (V, E)$. We now state as follows the decision version of the selection problem with vertex probabilities:

Problem: The decision version of SPvertex: DecSPvertex.

Instance: A simple, undirected graph $G = (V, E)$, a success probability $p(v)$, $0 \le p(v) \le 1$, for each $v \in V$, and numbers B and Z.

Question: Does there exist an edge set $E^* \subseteq E$ such that $|E^*| \le B$ and $\mathbb{E}((V, E^*), p) \ge Z$?

We state the following result.

Theorem 2. *DecSPvertex is NP-complete.*

3.1.3. Selection Problem with Explicit Scenarios. We finally define the decision version of the selection problem associated with a subset of scenarios. Given a compatibility graph $G = (V, E)$ with $K = 2$ and $L = 0$, we denote by $z(G)$ the optimal value of this KEP-instance, that is, the size of a maximum matching on G.

Problem: The decision version of SPscen: DecSPscen

Instance: A simple, undirected graph $G = (V, E)$, a collection of t edge sets $E_s \subseteq E$ $(1 \le s \le t)$, numbers B and Z.

Question: Does there exist an edge set $E^* \subseteq E$ such that $|E^*| \le B$ and $\sum_{s=1}^{t} z(V, E_s \cap E^*) \ge Z$?

In the statement of DecSPscen, the edge sets $E_1, \ldots, E_t$ represent t scenarios. The last inequality in the question implicitly assumes that all scenarios are equally likely, that is, each scenario has a probability $\frac{1}{t}$ of actually occurring.

We have the following statement.

Theorem 3. *DecSPscen is NP-complete.*

3.2. An Integer Programming Formulation of the Selection Problem

In order to model the selection problem as a two-stage programming problem, we introduce a first-stage binary variable $\beta_{i,j}$ for each arc $(i,j) \in A$, with the interpretation that $\beta_{i,j} = 1$ if and only if arc (i, j) is selected to undergo a crossmatch test. For a vector $\boldsymbol{\beta} \in \{0,1\}^{|A|}$ and for a scenario $A_s \in \mathcal{S}$, the set of arcs R that pass the crossmatch tests, as introduced in Section 2.3, is exactly $R = R_{s,\beta} = \{(i,j) \in A_s : \beta_{i,j} = 1\}$: in other words, $G_{s,\beta} = (V, R_{s,\beta})$ is the graph on which the recourse step will be performed by solving a deterministic KEP problem.

Let us denote by $\mathbf{x}_s \in \{0,1\}^{N_s}$ the vector of second-stage binary variables (of appropriate dimension) used to represent a solution to the KEP on the graph $G_s = (V, A_s)$, $s \in S$; we refer to these variables as *scenario* variables. Then, we can extend the KEP formulation (1)–(3) to obtain a generic two-stage integer programming formulation of the selection problem:

$$f_{\mathcal{S}} = \max \sum_{s \in \mathcal{S}} q_s \sum_{\ell=1}^{N_s} a_{s,\ell} x_{s,\ell}, \tag{7}$$

$$\text{subject to} \sum_{(i,j) \in A} \beta_{i,j} \le B, \tag{8}$$

$$\mathbf{x}_s \in P(G_{s,\beta}) \qquad \forall s \in \mathcal{S}, \tag{9}$$

$$\mathbf{x}_s \in \{0,1\}^{N_s} \qquad \forall s \in \mathcal{S}, \tag{10}$$

$$\beta_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{11}$$

The objective function (7) expresses the expected value of a solution of the selection problem by weighing, for each possible scenario s, the quality of the solution $\mathbf{x}_s$ by the probability of occurrence of scenario s. Constraint (8) ensures that at most B arcs are selected for crossmatch tests. Next, Constraints (9) link the selection of arcs to the choice of transplants to be performed: namely, they ensure that a solution x_s selected for scenario $s \in S$ defines a feasible set of transplants among the arcs that have been submitted to the crossmatch tests (as defined by $\boldsymbol{\beta}$) and that have passed them successfully (as expressed by scenario s).

The exact expression of Constraints (9) depends on the formulation adopted for KEP. It is usually easy to model (9) by simply adding to the formulation of $P(V, A_s)$ a collection of constraints of the form

$$\sum_{\ell \in Q(s,i,j)} x_{s,\ell} \le \beta_{i,j} \qquad \forall s \in \mathcal{S},\ \forall (i,j) \in A_s, \tag{12}$$

where the set $Q(s,i,j)$ is an appropriate set of indices, forcing some variables $x_{s,\ell}$ to be zero when $\beta_{i,j}$ is zero. In Online Appendix B, we explicitly state the formulation of the selection problem associated with the PIE formulation of KEP. Similar constraints can be used in conjunction with other formulations of KEP.

3.3. Selection Problem for a Restricted Subset of Scenarios

Computationally, the selection problem poses new challenges when compared with the recourse models discussed in Section 2.3. In particular, in approaches that solve those recourse models, all relevant cycles c

or subsets U can be explicitly generated if their sizes are sufficiently small. Moreover, the computation of the parameters w'_c or w_U (the expected number of transplants in a subgraph induced by a cycle c or by a subset U) is also facilitated by the fact that the subgraphs under consideration are small (Pedroso 2014, Klimentova et al. 2016). In contrast with these observations, taking into account $|\mathcal{S}| = 2^{|A|}$ different scenarios in model (7)–(11) results in a very large number of decision variables and of constraints.

In order to obtain a tractable model, we use the so-called sample average approximation (SAA) technique. SAA is a popular technique that is used to approximate an expected value function by a sample average function, thereby alleviating the computational burden involved with a huge number of scenarios while remaining accurate. We refer to Homem-de-Mello and Bayraksan (2014) for a survey describing the use of SAA in many diverse situations, and we refer to Kleywegt et al. (2001) for an in-depth treatment of this approach. Applying SAA to our setting allows us to restrict our attention to a subset of the scenarios. More precisely, we replace, in (7)–(11), the set of scenarios $\mathcal{S}$ by the set $S \subseteq \mathcal{S}$ where each scenario $s \in S$ is weighted by a factor $1/|S|$ in the objective function. Thus, we have constructed the following model:

$$f_S = \max \sum_{s \in S} \frac{1}{|S|} \sum_{\ell=1}^{N_s} a_{s,\ell} x_{s,\ell}, \tag{13}$$

$$\text{subject to} \sum_{(i,j) \in A} \beta_{i,j} \leq B, \tag{14}$$

$$\mathbf{x}_s \in P(G_{s,\beta}) \qquad \forall s \in S, \tag{15}$$

$$\mathbf{x}_s \in \{0,1\}^{N_s} \qquad \forall s \in S, \tag{16}$$

$$\beta_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{17}$$

The model (13)–(17) is referred to as the *restricted selection problem* associated with the subset S of scenarios.

By using SAA, each scenario $s \in \mathcal{S}$ has probability q_s to be included in S; we expect $\mathbb{E}(f_S)$ to be a good approximation of $f_{\mathcal{S}}$, at least when the number of scenarios is sufficiently large. Kleywegt et al. (2001) actually show that with probability one, when the sample size goes to infinity, $\mathbb{E}(f_S)$ converges to $f_{\mathcal{S}}$, and the optimal solutions of the restricted problem (13)–(17) are optimal for the complete problem (7)–(11) (see proposition 2.1 in Kleywegt et al. 2001). However, for a fixed sample size, the expected optimal value of (13)–(17) overestimates the true optimal value, that is, $f_{\mathcal{S}} \leq \mathbb{E}(f_S)$, and hence $\mathbb{E}(f_S)$ is a biased estimator (Kleywegt et al. 2001). We briefly return to this point in the discussion of our computational experiments (Section 7).

Constraints (15) and (16) model the deterministic KEP over the subgraph $G_{s,\beta}$ associated with the selection $\boldsymbol{\beta}$ and with scenario s. Extensive computational experiments by Dickerson et al. (2016) have shown that the continuous relaxation of position-indexed formulations of the deterministic version of KEP are extremely tight in practice (the Linear Programming (LP) bound was equal to the optimal Integer Programming (IP) value for more than 90% of the large-scale real-world instances that they tested). Our own preliminary experiments confirmed that most formulations of the (deterministic) KEP have very tight linear relaxations. These observations suggest that replacing the binary scenario variables (say, the generic variables x_s in Formulation (7)–(11) of the selection problem) by their continuous relaxation is unlikely to have a big impact on the arcs that are picked for testing, that is, on the optimal value of the $\beta_{i,j}$ variables (see Section 7 for an experimental assessment of this assumption). The problem obtained on replacing (16) by

$$0 \leq x_s \leq 1 \qquad \forall s \in S \tag{18}$$

is called the *relaxed restricted selection problem*.

For both the restricted and the relaxed restricted selection problem, it is crucial to understand that, for every subset of scenarios, any values of the $\beta_{i,j}$-variables satisfying Constraint (14) provide a feasible selection and can be implemented as a first-stage solution of the stochastic KEP.

4. Investigating the IP Formulation of the Selection Problem

Let us now turn to algorithms for the solution of the selection problem (over the complete set $\mathcal{S}$ or over an arbitrary subset S of scenarios: the same discussion applies in both cases, *mutatis mutandis*). Because the vast majority of variables in any formulation of the selection problem are scenario variables, relaxing them decreases the number of binary variables by an order of magnitude. Moreover, when the scenario variables are relaxed, Benders decomposition becomes a natural solution approach for the resulting mixed-integer program: indeed, after fixing the $\beta_{i,j}$ variables to either zero or one in (7)–(11), the problem breaks down into smaller independent subproblems, where each subproblem is the linear relaxation of KEP over the subgraph $G_{s,\beta}$ defined by $\boldsymbol{\beta}$ and by a specific scenario s.

More precisely, relaxing the scenario variables in the selection problem and using Benders decomposition, we obtain the *master problem*

$$\max \sum_{s \in \mathcal{S}} q_s z_s(\boldsymbol{\beta}), \tag{19}$$

$$\text{subject to} \sum_{(i,j) \in A} \beta_{i,j} \leq B, \tag{20}$$

$$\beta_{i,j} \in \{0,1\} \qquad \forall (i,j) \in A, \tag{21}$$

where $z_s(\boldsymbol{\beta})$ is the optimal value of the following *subproblem*, for each $s \in \mathcal{S}$:

$$z_s(\boldsymbol{\beta}) = \max \sum_{\ell=1}^{N_s} a_{s,\ell} x_{s,\ell}, \tag{22}$$

$$\text{subject to } \mathbf{x}_s \in P(G_{s,\beta}), \tag{23}$$

$$0 \le x_{s,\ell} \le 1 \qquad \forall\, \ell = 1, \ldots, N_s. \tag{24}$$

Without going into details (we refer to Rahmaniani et al. 2017 for a discussion of Benders decomposition), let us simply mention here that, with Constraints (12), solving the dual problem of (22)–(24) yields valid inequalities of the generic form

$$\sum_{(i,j)\in A_s} u^t_{i,j}\beta_{i,j} + v^t \ge z_s(\boldsymbol{\beta}) \qquad \forall\, t = 1, \ldots, m, \tag{25}$$

where $u^t_{i,j}$ and v^t are coefficients derived from the extreme points of the dual region. These inequalities can be added to (19)–(21) to obtain the following reformulation of the master problem:

$$\max \sum_{s\in\mathcal{S}} q_s z_s, \tag{26}$$

$$\text{subject to } \sum_{(i,j)\in A} \beta_{i,j} \le B, \tag{27}$$

$$\sum_{(i,j)\in A_s} u^t_{i,j}\beta_{i,j} + v^t \ge z_s \qquad \forall\, t = 1, \ldots, m,\ \forall\, s \in \mathcal{S}. \tag{28}$$

$$\beta_{i,j} \in \{0,1\} \qquad \forall\, (i,j) \in A. \tag{29}$$

Because there is an exponential number of constraints (28), adding them all at once in the model is not efficient. Benders decomposition approaches therefore typically work iteratively, starting at each iteration with a *restricted master problem* of the form (26)–(29), but which only includes a subset of Constraints (28). The value of the β variables in an (optimal) solution of this restricted master problem is then used to formulate the dual of the subproblem (22)–(24) for each scenario $s \in \mathcal{S}$ and to generate new valid inequalities that can be added in the restricted master problem. In our computational experiments, this iterative process is automatically managed by a commercial mixed integer program (MIP) solver (CPLEX). Before handing the model to the solver, however, we found it useful to strengthen the formulation as explained in the next section.

4.1. Strengthening the Formulation

In an optimal solution of the selection problem, each selected arc must be part either of a cycle of length at most K, or of a chain of length at most L: indeed, only such arcs can potentially be used for transplants. In the Benders decomposition of the problem, however, this information is lost to the master problem. In order to palliate this weakness, we add to (7)–(11) (or to (13)–(17)) a list of constraints that force the selected arcs to be part of a cycle or chain. In this way, the solutions of the restricted master problem will display more features of optimal solutions of the full selection problem, even in early iterations of the generation of Benders cuts.

To achieve this goal, we rely on a formulation related to the position-indexed edge formulation of Dickerson et al. (2016) (see Online Appendix B). Namely, for each vertex ℓ in the graph, a graph copy G^ℓ is created, where G^ℓ is the graph G from which all vertices i with $i < \ell$ have been deleted. Then, new binary variables are created and constrained in such a way, that in graph copy $i < \ell$, we choose the arc leaving vertex ℓ to be in the first position in a cycle or chain. Namely, variable $\phi^\ell_{i,j,k}$ is equal to one only if the arc (i, j) is in the kth position of some cycle in graph copy G^ℓ, for all $i,j,\ell \in V, i,j \ge \ell, (i,j) \in A$, and $k = 1, \ldots, K$. As in Dickerson et al. (2016), the variables $\phi^\ell_{i,j,k}$ can be defined only for $k \in \mathcal{K}(i,j,\ell)$, where the set $\mathcal{K}(i,j,\ell)$ contains those positions in which arc (i, j) can potentially occur in a cycle of G^ℓ. In particular, $\mathcal{K}(i,j,\ell) \subseteq \{1, \ldots, K\}$, $1 \in \mathcal{K}(i,j,\ell)$ only if $i = \ell$, and $K \in \mathcal{K}(i,j,\ell)$ only if $j = \ell$. Similarly, we define variables $\omega_{i,j,k}$ to be equal to one only if the arc (i, j) is in the kth position of some chain, for all $(i,j) \in A$, and $k = 1, \ldots, L$. Here again, k can be restricted to a subset $\mathcal{K}(i,j)$ of feasible positions, where $\mathcal{K}(i,j) \subseteq \{1, \ldots, L\}$ and $1 \in \mathcal{K}(i,j)$ only if i is a nondirected donor.

The following constraints can now be added to ensure that each selected arc is part of a cycle or chain of selected arcs:

$$\beta_{i,j} - \sum_{\ell\in V} \sum_{k\in\mathcal{K}(i,j,\ell)} \phi^\ell_{i,j,k} - \sum_{k\in\mathcal{K}(i,j)} \omega_{i,j,k} \le 0 \qquad \forall\, (i,j) \in A, \tag{30}$$

$$\phi^\ell_{i,j,k} - \beta_{i,j} \le 0 \quad \forall \ell \in V, (i,j) \in A, k \in \mathcal{K}(i,j,\ell), \tag{31}$$

$$\omega_{i,j,k} - \beta_{i,j} \le 0 \qquad \forall\, (i,j) \in A, k \in \mathcal{K}(i,j), \tag{32}$$

$$\phi^\ell_{i,j,k} - \sum_{(h,i)\in A:(k-1)\in\mathcal{K}(h,i,\ell)} \phi^\ell_{h,i,k-1} \le 0 \qquad \forall\, \ell \in V, (i,j) \in A, k \in \mathcal{K}(i,j,\ell), k > 1, \tag{33}$$

$$\phi^\ell_{i,j,k} - \sum_{(j,h)\in A:(k+1)\in\mathcal{K}(j,h,\ell)} \phi^\ell_{j,h,k+1} \le 0 \quad \forall \ell \in V, (i,j) \in A : j \ne \ell, k \in \mathcal{K}(i,j,\ell), k < K, \tag{34}$$

$$\omega_{i,j,k} - \sum_{(h,i)\in A:(k-1)\in\mathcal{K}(h,i)} \omega_{h,i,k-1} \le 0 \qquad \forall\, (i,j) \in A, k \in \mathcal{K}(i,j), k > 1. \tag{35}$$

These constraints can be interpreted as follows. Constraints (30)–(32) enforce that an arc can only be

selected if it is part of a chain or cycle, whose arcs are also selected. Constraints (33) enforce that, if arc (i, j) is in position k in some cycle of G^ℓ, then there must be a preceding arc in position $k-1$, unless $k = 1$; when $k = 1$, then there is no preceding arc, but the variables $\phi^\ell_{i,j,1}$ are constructed in such a way that i must necessarily be equal to ℓ. Similarly, Constraints (34) enforce that (i, j) must have a succeeding arc, unless $j = \ell$ (which ensures that the cycle is completed). Finally, constraints (35) enforce that if an arc (i, j) is chosen in the kth position ($k > 1$) of a chain, then another arc must be chosen in the preceding position. If $k = 1$, there is no preceding position, but the variables are constructed in such a way that the starting vertex of the arc corresponds to a nondirected donor. In this way, the variables $\omega_{i,j,k}$ encode chains in the graph G.

In (30)–(35), contrary to the PIE formulation of Dickerson et al. (2016), we allow selected cycles and chains to overlap, so that several in- or outgoing selected arcs may be incident to a same vertex. This is justified by the fact that, in the recourse step defined in Section 2.3, different cycles or chains may be chosen depending on the observed scenario.

5. Experimental Setting

In the next sections, we discuss the quality of the outcomes of the selection model and the efficiency of the approaches that we have tested. We start here with a description of the experimental setting.

For our experiments, we generate kidney exchange graphs using the random generator described in Saidman et al. (2006). We consider three different graph sizes, consisting, respectively, of $n = 25$ patient-donor pairs, $n = 50$ patient-donor pairs, and $n = 25$ patient-donor pairs plus one nondirected donor. For each of the sizes 25, 50, and 25 + 1, we generate 40 graphs, divided into four groups of 10 graphs, with vertex success rates p equal to 0.8, 0.6, 0.4, and 0.2 in these four groups, respectively. For size 50, we added a fifth set of 10 instances, where each vertex success rate is drawn from a uniform distribution between 0.2 and 0.8, independently for each vertex. (The assumption of equal success probability for all vertices is common in the KEP literature; see Alvelos et al. 2015 and Dickerson et al. 2019). Although it may seem unrealistic, there is an ethical argument for postulating equal probabilities. Dickerson et al. (2019), for instance, explicitly argue that equal failure probabilities are useful to protect more vulnerable patients. On the other hand, the so-called *heterogeneous* instances, with different success probabilities for all vertices, provide insight into the possible gains in the number of expected transplants, when success rates within the population can differ significantly.)

The previous graphs form the basis for all instances.[1] Moreover, in each instance, we limit the cycle length K to two, three, or four, and we limit the maximum chain length L to either zero or three (because these are the most common lengths in real-life kidney exchanges). To complete the instances, we still must specify the upper bound B on the number of selected arcs. In order to ensure a fair comparison with recourse approaches from literature, we set B equal to the number of arcs selected in a solution found by one of the recourse models (either subset, or internal, or no recourse model, depending on the experiment) on the same graph for a similar configuration. Finally, all instances feature either $|S| = 50$, 100, 250, or 500 random scenarios in Formulation (13)–(17) of the restricted selection problem. To be able to compare the solution quality and the computation times of the different algorithmic approaches, each approach uses the exact same scenario set S.

We solve the restricted selection problem using the PIE formulation (Online Appendix B) by the following four solution approaches:

1. BC-IP: Solve the restricted selection IP problem (13)–(17) using the standard CPLEX branch-and-cut solver.
2. BC-MIP: Solve the relaxed restricted MIP selection problem (13)–(15), (17), and (18) using the standard CPLEX branch-and-cut solver.
3. BD-MIP: Solve the relaxed restricted selection MIP problem (13)–(15), (17), and (18) by Benders decomposition.
4. BD-MIP+: Solve the relaxed restricted selection MIP problem (13)–(15), (17), and (18), augmented by Constraints (30)–(35), by Benders decomposition.

For each solution approach, the corresponding model is solved using CPLEX 12.8, called through the Concert API. For the two Benders-based approaches, we rely on CPLEX's built-in Benders functionality. All instances were run with a time limit of two hours on four SandyBridge 2.6-GHz CPUs with 8 GB of memory.

Notice that BC-IP attempts to solve the IP formulation of the problem, whereas BC-MIP, BD-MIP, and BD-MIP+ are solution approaches for different MIP relaxations of the problem. Yet a common feature of solutions of each of these approaches is that they select a set of arcs, namely the set of arcs (i, j) for which $\beta_{i,j} = 1$ in the respective formulation, and hence, they yield a feasible solution $\boldsymbol{\beta}$ of the selection problem. In the sequel, when we use the word "solution," we actually refer to this set of arcs. In case an instance was not completely solved within two hours, the best incumbent solution was evaluated (see Section 8 for more details about the instances solved).

To assess the quality of a solution $\boldsymbol{\beta}$ obtained by any method, we must evaluate the (true) expected number of transplants associated with $\boldsymbol{\beta}$. (Observe that the objective function value of the model is not exactly equal to the expected number of transplants, in view of the

reliance of SAA on restricted subsets of scenarios, and of our use of linearly relaxed formulations.) These evaluations were performed by extensive ex post computations, in one of two ways. For some of the instances, the expected number of transplants was computed by a modified version of the algorithm used by Klimentova et al. (2016) for the evaluation of subsets. This procedure allows us to evaluate exactly (ex post) the quality of a solution $\boldsymbol{\beta}$. However, because of the presence of large connected components in the graphs that need to be evaluated, this procedure sometimes turned out to be computationally costly. Therefore, for most instances, the solution was evaluated by a Monte Carlo procedure. Namely, given the selection solution $\boldsymbol{\beta}$ for these instances, we generated a large number of random scenarios, and we used these scenarios to compute an estimate of the expected number of transplants.

6. Outcomes of the Selection Model vs. Other Approaches

In this section, we compare the quality of the solutions of the selection model found by our approaches, with the quality of the solutions delivered by internal recourse, subset recourse, and no recourse (see Section 2.3 for their description). In Section 6.1, we investigate results compared with recourse approaches in the literature, whereas Section 6.2 discusses the no-recourse failure aware matching. In Section 6.3 we investigate the scaling of our observations to larger datasets.

6.1. Recourse Approaches

In this section, we compare the outcomes of the selection model with the outcomes found by the recourse approaches, namely, internal recourse (IR) and subset recourse (SR). Table 1 gives the expected number of transplants found by IR and by SR, together with the expected number of transplants found by solving the corresponding selection problem. For the selection problem, we report on the results obtained by BD-MIP on the relaxed restricted formulation (all methods actually provide very similar results for these instances of the selection problem, as we will see in the next sections). In particular, Table 1 displays the results for instances with $N = 25, K = 3, L = 0$, with $N = 25, K = 4, L = 0$, and with n = 25 + 1 (directed donor), $K = 3, L = 3$. The entries in the columns called IR, SR, and the corresponding column Selection give the average number of transplants realized by the corresponding approach (recall that each entry is an average over 10 instances). The column labeled % Increase displays the improvement provided by the selection model over the corresponding recourse model. The number of allowed crossmatch tests (given in the columns called B) is the number of arcs tested in the

Table 1. Comparison of the Expected Number of Transplants Between (i) Internal Recourse (IR) and Selection and (ii) Subset Recourse (SR) and Selection (BD-MIP, 500 Scenarios)

p	IR	Selection	% Increase	B	SR	Selection	% Increase	B
				$N = 25, K = 3, L = 0$				
0.8	6.12	6.14	0.3%	12.5	6.78	6.91	1.9%	19.1
0.6	3.90	4.02	3.1%	14.6	4.60	4.64	0.9%	22.5
0.4	1.83	1.94	6.0%	14.7	2.32	2.39	3.0%	22.9
0.2	0.45	0.50	11.1%	14.5	0.65	0.69	6.2%	22.9
				$N = 25, K = 4, L = 0$				
0.8	6.69	6.91	3.3%	16.5	6.99	7.08	1.3%	19.5
0.6	4.27	4.47	4.7%	19.8	4.62	4.68	1.3%	22.3
0.4	2.14	2.31	7.9%	20.8	2.33	2.38	2.1%	22.6
0.2	0.56	0.63	12.5%	19.8	0.65	0.69	6.2%	23.1
				$N = 25 + 1, K = 3, L = 3$				
0.8	7.52	7.66	1.9%	16.1	8.16	8.30	1.7%	23.0
0.6	4.89	5.13	4.9%	19.6	5.73	5.82	1.6%	28.3
0.4	1.88	2.05	9.0%	16.0	2.41	2.46	2.1%	23.5
0.2	0.53	0.59	11.3%	17.1	0.70	0.72	2.9%	25.3

outcome of the solution found using internal or subset recourse. Indeed, recall that this number of tests is not a priori bounded in a recourse approach, while it must be when formulating the selection problem. Hence, in order to allow for a fair comparison of the outcomes, we opted to set B equal, in each instance, to the number of arcs used in the solution computed by the recourse approach under consideration.

Let us now discuss the results displayed in Table 1. Of course, for each of the approaches we find that the larger p, the higher the number of expected transplants. More importantly, we conclude from the results that solutions found by the selection model dominate the solutions found by the recourse approaches for all values of p. In fact, as the value of p becomes smaller, the improvement in quality of the solutions found by the selection model gets larger. Especially compared with internal recourse, the selection model brings significant improvements, ranging up to 12%, at low vertex success probabilities. A closer look at the solutions suggests that this is mostly because of the inability of internal recourse to use overlapping two cycles, which, at low success rates, provide many more expected transplants per tested arc than longer cycles.

6.2. No Recourse

A key factor in explaining the improvements of the selection model over the outcome of recourse approaches lies in the number of tests used by the selection model. Indeed, when we compare the selection model with the no recourse setting (NR), the quality of the solutions obtained by these approaches is found to be quite similar, as shown in Table 2. This is because of the relatively small number of arcs that the

no-recourse approach uses, leading to small values of B that prohibit the selection problem from finding better solutions (compare the value of B in Tables 1 and 2). In these unnecessarily constrained cases, randomness in the scenario set may even lead the selection model to select some overlapping two cycles, which are inferior to nonoverlapping two cycles chosen by the no-recourse model.

6.3. Larger Data Sets

The scaling of the selection method to larger data sets is an important consideration. The two main questions are whether the improvement over the recourse approaches carries over to larger data sets and whether the selection model can be solved for larger data sets. The two questions are clearly linked, because optimal solutions must be found in order to evaluate the full potential of the selection model. For 50 patient-donor pairs, as we will discuss in Section 8.4, we cannot solve all instances to optimality within eight hours. However, evaluating the best solutions found within the time limit suggests a positive answer to the first question. For example, in Table 3, we compare the best solutions found by the selection method against those found by subset recourse. The last column of the table shows the average optimality gap for the selection model after eight hours of computing time. For all classes of instances, the conclusions drawn for smaller instances still apply, and the solutions of the selection model dominate those obtained by subset recourse, although many instances have not been solved to optimality for the selection model. The improvement over subset recourse is especially marked for instances with $p = 0.2$, but also for heterogeneous instances.

Table 2. Comparison of the Expected Number of Transplants Between No Recourse and Selection (BD-MIP, 500 Scenarios)

p	No recourse	Selection	% Increase	B
		$N = 25, K = 3, L = 0$		
0.8	5.55	5.56	0.2%	10.0
0.6	3.16	3.16	0.0%	8.9
0.4	1.52	1.52	0.0%	9.7
0.2	0.35	0.33	−5.3%	8.9
		$N = 25, K = 4, L = 0$		
0.8	5.63	5.76	2.3%	10.7
0.6	3.16	3.16	0.0%	8.9
0.4	1.52	1.52	0.0%	9.7
0.2	0.35	0.33	−5.3%	8.9
		$N = 25 + 1, K = 3, L = 3$		
0.8	6.75	6.74	−0.1%	11.9
0.6	4.00	4.00	0.0%	11.8
0.4	1.49	1.53	2.7%	10.1
0.2	0.37	0.41	10.8%	10.8

Table 3. Comparison of Expected Number of Transplants Between Subset Recourse and Selection (BD-MIP+, $N = 50$, $K = 3, L = 0$, 500 Scenarios, Eight-Hour Time Limit)

p	SR	Selection	% Increase	B	Optimality gap
0.8	17.76	17.93	0.95%	52.6	2.6%
0.6	11.72	11.75	0.31%	56.5	5.0%
0.4	6.08	6.09	0.18%	59.6	5.8%
0.2	1.75	1.85	5.81%	60.2	0%
Heterogeneous	8.32	8.73	4.89%	54.0	3.7%

These results suggest that also for larger pool sizes, the selection model can increase the expected number of transplants, despite the difficulty of solving the model.

7. Impact of Restricted and Relaxed Formulations

In this section, we investigate the impact on the quality of a solution when (i) we restrict the number of scenarios and (ii) we relax the integrality of the scenario variables

In Section 3.3, we introduced the SAA restricted selection problem, where we limit ourselves to a subset S of all possible scenarios; the resulting problem is solved by approach BC-IP. Next, we suggested the relaxed restricted selection problem, which arises when we relax the integrality of the scenario variables and we can use in the three solution approaches BC-MIP, BD-MIP, and BD-MIP+. Both the restriction and the relaxation of the original selection problem presumably simplify the computational problem but may result in solutions of lower quality compared with the optimum f_S. Recall that the quality of a solution is measured by the expected number of transplants that can be realized in a graph that has as an arc-set those arcs (i) that are selected and (ii) that pass the crossmatch test as specified by given probabilities. Optimizing over a restricted subset of scenarios may lead to overfitting, that is, to the selection of (combinations of) arcs that are disproportionately successful in the chosen scenarios compared with the full range of scenarios. On the other hand, relaxing the scenario variables may entail the selection of arcs for which good fractional KEP solutions exist for all scenarios but no good integer solution.

Table 4 reports the results obtained on graphs with $n = 25$ patient-donor pairs and various values of $p, |S|, K, L$.

Each entry provides an estimate (averaged over 10 instances) of the expected number of transplants associated with the optimal solution obtained by the approach indicated in the column header (either BC-IP or BD-MIP+).

7.1. Restricting the Number of Scenarios

From Table 4, we see that the effect of varying the number of scenarios depends mostly on the success

Table 4. Estimate of the Expected Number of Transplants (Average over 10 Instances), $N = 25$

		$K = 2, L = 0$		$K = 3, L = 0$		$K = 4, L = 0$		$K = 3, L = 3$	
p	$\|S\|$	BC-IP	BD-MIP+	BC-IP	BD-MIP+	BC-IP	BD-MIP+	BC-IP	BD-MIP+
0.8	50	5.37	5.37	6.85	6.80	7.08	7.08	8.21	8.27
	100	5.38	5.38	6.88	6.88	7.16	7.06	8.29	8.28
	250	5.38	5.36	6.90	6.87	7.17	7.08	8.26	8.31
	500	5.38	5.38	6.91	6.91	7.14	7.08	8.31	8.30
0.6	50	4.06	4.06	4.54	4.55	4.52	4.49	5.74	5.73
	100	4.07	4.06	4.57	4.58	4.58	4.58	5.78	5.78
	250	4.07	4.07	4.62	4.63	4.65	4.65	5.79	5.82
	500	4.08	4.08	4.64	4.64	4.67	4.68	5.85	5.82
0.4	50	2.19	2.19	2.27	2.28	2.23	2.25	2.30	2.28
	100	2.21	2.21	2.32	2.33	2.31	2.31	2.39	2.39
	250	2.21	2.21	2.38	2.38	2.37	2.36	2.44	2.43
	500	2.22	2.22	2.39	2.39	2.38	2.38	2.46	2.46
0.2	50	0.61	0.61	0.59	0.56	0.55	0.54	0.56	0.60
	100	0.63	0.63	0.64	0.64	0.63	0.63	0.67	0.67
	250	0.63	0.63	0.68	0.68	0.68	0.69	0.71	0.71
	500	0.64	0.64	0.69	0.69	0.69	0.69	0.72	0.72

rate and to a lesser extent on the maximum cycle/chain length. When $K = 2$ or when $p = 0.8$, the number of scenarios used hardly affects the quality of the solution. However, for $K = 3$ and $L = 0$, enlarging the number of scenarios from 50 to 500 increases the number of expected transplants of the (near-)optimal solution by less than 1% for the instances with 80% success rate, whereas it increases the solution quality by 17% for the instances with 20% success rate. Therefore, it appears that for smaller success rates and for larger values of K and L that increasing the number of scenarios matters. The influence of the success rate is further illustrated by the difference between the objective value of the restricted selection problem (i.e., the expected number of transplants estimated over a limited subset S of scenarios), and the ex post evaluation over a much larger set. For instances with $K = 3$ and $L = 0$, 50 scenarios, and 20% success rate, the accurate ex post evaluation of the objective value of the restricted selection problem (BC-IP) equals 0.59 (Table 4). However, the objective function value given by the optimal solution of BC-IP (again averaged over 10 instances) equals 0.82 (this value is not shown in Table 4), which is an overestimation of 44%. By comparison, for instances with $K = 3$, $L = 0$, 50 scenarios, and 80% success rate, the average objective function value is 7.04, an overestimation of only 3% over the more accurate value 6.85.

7.2. Relaxing the Scenario Variables

The impact of relaxing the scenario variables can be measured by the difference between each value in column IP and the corresponding value in column MIP in Table 4. The results clearly show that relaxing the scenario variables has little to no influence on the quality of solutions for any set of values we consider for K and L. This is fully coherent with the observations of Dickerson et al. (2016), which motivated the consideration of the relaxed model, as discussed in Section 3.3.

We point out that, although instances of the KEP with $K = 2, L = 0$ are solvable in polynomial time, the LP-relaxations of the corresponding formulations need not be integral, and it is therefore not guaranteed that the values in columns IP and MIP are close. However, Table 4 shows that, in general, the value of the solution found by solving the MIP relaxation lies quite close to the value of the solution of the IP formulation. This suggests the relaxation does not come at the cost of solution quality.

8. Computational Efficiency

In this section, we focus on the computational efficiency of different solution approaches, and we investigate how computing times depend on various parameters.

In a first experiment, we compare all four solution approaches on graphs of size 25, with maximum cycle length 3 and no chains (Section 8.1). In a second experiment, we investigate the impact of the maximum cycle length, by comparing the computation times for cycle lengths 2, 3, and 4 on the same graphs (Section 8.2). In a third set of experiments, we allow the use of chains in graphs of size 25 + 1 (Section 8.3). Finally, we report on our experience with larger graphs in Section 8.4.

8.1. Comparison of the Running Times of the Solution Approaches

In this first experiment, we ran all instances with each solution method (BC-IP, BC-MIP, BD-MIP, BD-MIP+) for cycle length $K = 3$.

Table 5. Average Computation Times (in Seconds) over 10 Instances for Each Solution Method, $N = 25, K = 3, L = 0$

p	$\|S\|$	BC-IP	BC-MIP	BD-MIP	BD-MIP+
0.8	50	15	22	518	33
	100	23	23	510	18
	250	149	314	1,124	91
	500	812*	813*	1,758*	232
0.6	50	12	13	14	8
	100	14	16	18	41
	250	37	48	761	45
	500	154	156	1,073	58
0.4	50	11	14	14	10
	100	11	12	12	13
	250	19	21	42	37
	500	31	31	172	36
0.2	50	1.6	4.8	3.0	3.6
	100	2.8	8.4	5.8	5.3
	250	10	10	8.6	7.4
	500	13	17	10	9.2

Notes. Each asterisk refers to one instance (out of 10) that did not finish within two hours. In such cases, unfinished instances are counted as taking 7,200 seconds.

It appears from the results in Table 5 that BD-MIP+ (relaxing the scenario variables, applying Benders decomposition and using Constraints (30)–(35)) generally leads to the lowest computation times, especially for harder instances involving a large number of scenarios or a large success probability. For small numbers of scenarios, we also note the good performance of BC-IP (somewhat surprisingly, in most cases, this approach is actually more efficient than its mixed-integer relaxed counterpart BC-MIP). However, as already mentioned in Section 7, we point out that the quality of the solutions decreases significantly with the number of scenarios. Standard Benders decomposition (BD-MIP) does not perform well except on the instances with low success rate. In fact, BD-MIP is generally even slower than BC-IP. On the other hand, adding Constraints (30)–(35) in BD-MIP+ has a very strong, positive impact on the computation time.

8.2. Impact of Cycle Length

In our second set of experiments, we solve the (relaxed) restricted selection problem for different values of K, that is, for different maximum cycle lengths. Average computation times of the two solution approaches BC-IP and BD-MIP+ are displayed in Table 6, with the shortest times in boldface for each combination of instance type and solution method.

From the results in Table 6, we see that the maximum cycle length has a strong influence on the total computation time. Especially in the instances with high success rates, computation times increase sharply with the cycle length. The results also mostly confirm the trend observed in Section 8.1, with BD-MIP+ performing better than BC-IP for the harder instances. Interestingly, however, BC-IP remains a competitive alternative when the number of scenarios is not too large.

8.3. Impact of Chains

In our third set of experiments, we investigate the impact of the presence of nondirected donors (thereby allowing chains) on the computation times. To set the upper bound on the number of selected arcs, we run subset recourse (see Section 2.3) on a modified version of the graph (because subset recourse was not originally developed to handle chains): namely, for each vertex associated with a patient-donor pair, we add an arc with weight 0 from that vertex to the vertex representing the nondirected donor (referred to as the NDD vertex). We furthermore allow cycles of different lengths if they start from the NDD vertex.

Table 7 shows that the presence of nondirected donors has a significant impact on computation times. Without a nondirected donor, most instances with cycle length 3 and 80% success rate are solved within two hours, even with 500 scenarios. In contrast, the addition of a single nondirected donor results in many instances that could not be solved for the harder configurations. When using BD-MIP+, 8 of 20 instances with 500 scenarios fail to finish within the time limit for the instances with 80% and 60% success rates; this happens for 5 of 20 instances when dealing with 250 scenarios. The density of the graph, and as a consequence, the upper bound on the number of

Table 6. Average Computation Times (in Seconds) over 10 Instances for $N = 25, K \in \{2, 3, 4\}, L = 0$

		$K = 2, L = 0$		$K = 3, L = 0$		$K = 4, L = 0$	
p	$\|S\|$	BC-IP	BD-MIP+	BC-IP	BD-MIP+	BC-IP	BD-MIP+
0.8	50	3.3	**0.2**	15	33	153	**117**
	100	3.6	**0.9**	23	**18**	401	**181**
	250	3.9	**3.4**	149	**91**	1,156*	**730**
	500	13	**9.0**	812*	**232**	2,149*	**1,025***
0.6	50	3.0	**0.2**	12	8.3	**29**	132
	100	5.0	**1.6**	**14**	41	**40**	52
	250	**6.5**	8.1	**37**	45	266	**228**
	500	**7.6**	25	154	**58**	1,502	**936**
0.4	50	3.6	**2.9**	11	**10**	**15**	20
	100	6.1	**2.6**	**11**	13	**12**	15
	250	**7.3**	9.4	**19**	37	**46**	121
	500	**8.4**	11	**31**	37	115	**71**
0.2	50	1.1	**0.1**	**1.6**	3.6	**2.2**	3.9
	100	4.1	**0.0**	**2.8**	5.3	**5.6**	7.0
	250	8.3	**2.3**	10	**7.4**	**11**	14
	500	11	**3.8**	13	**9.2**	17	**14**

Notes. Each asterisk refers to one instance (out of 10) that did not finish within two hours. In such cases, unfinished instances are counted as taking 7,200 seconds.

Table 7. Average Computation Time for Different Solution Methods over 10 Instances, $N = 25 + 1$ Nondirected Donor, $K = 3, L = 3$

p	$\|S\|$	BC-IP	BD-MIP	BD-MIP+
0.8	50	106	69	**57**
	100	196	285	**122**
	250	(9) 2,389	(9) 1,886	(9) **1,748**
	500	(4) 4,385	(7) **3,283**	(6) 3,296
0.6	50	1,034	1138	**774**
	100	(7) 2,621	**2,080**	(7) 2,976
	250	(4) 4,354	(8) **3,379**	(6) 3,853
	500	(4) 4,409	(4) 4,756	(6) **4,059**
0.4	50	**9.6**	17.8	28.1
	100	**14**	20	21
	250	451	**438**	742
	500	785	**739**	781
0.2	50	1.3	1.0	**0.3**
	100	**1.1**	**1.1**	2.4
	250	12	9.5	**8.4**
	500	16	13	**11**

Notes. Numbers in parentheses indicate the number of instances completed in under two hours. Unfinished instances are counted as taking 7,200 seconds.

selected arcs derived from the subset recourse solution do play an important role. Although 4 of 10 instances with 80% success rate and 500 scenarios failed to finish in two hours, 3 of these instances finished in under 20 seconds. These two groups of instances contain 182 and 122 arcs on average, respectively. The difficulty of the instances with 60% success rate is also apparently increasing with the density of the graphs.

Comparing the different solution approaches, we observe that BD-MIP and BD-MIP+ consistently outperform BC-IP for the harder instances with high success rates. For the easier instances, results are more mixed. Unlike in the case of cycle-only instances, BD-MIP is competitive with the other approaches.

8.4. Larger Datasets

We shortly comment on our computational experience with graphs of size 50. Instances of this size often prove too large to solve within the time limit, or without running out of memory, by any approach. Table 8 displays the results obtained in 8 hours of computing time with BC-IP and with BD-MIP+. For 80% success rate and 50 scenarios, BD-MIP+ solves 7 of 10 instances to optimality within the time limit, but this ratio falls to 4 of 10 for 500 scenarios. BC-IP performs slightly worse on these instances. For all other classes of instances, results are more mixed. The efficiency of the algorithms clearly decreases when the number of scenarios increases. Nevertheless, we are able to solve to optimality (by either method) all instances with 20% success rate and 100 scenarios; even with 500 scenarios, 9 of 10 instances are solved to optimality. We finally note that both methods obtain solutions of similar quality. BD-MIP+ is more memory intensive, whereas BC-IP only fails to solve to optimality because of the time limit, and BD-MIP+ is prone to running out of memory.

Table 8. Instances Solved to Optimality in Under Eight Hours by Success Rate and Number of Scenarios ($N = 50$)

		BC-IP				BD-MIP+			
p	$\|S\|$	50	100	250	500	50	100	250	500
0.8		6	5	2	1	7	5	6	4
0.6		8	3	3	2	6	5	5	1
0.4		9	8	4	2	7	5	2	2
0.2		10	10	10	9	10	10	9	9
Heterogeneous		7	6	5	4	5	8	9	5

9. Conclusions

Selecting the right patient-donor pairs for crossmatching in kidney exchange programs is an important and challenging problem. In this paper, we introduced a new model for the optimization of kidney exchanges under stochastic failures. We formulated the resulting problem as a two-stage stochastic optimization problem. In terms of the expected number of transplants, our model compares favorably with recourse models in the literature. Hence, institutions that are responsible for running a kidney exchange program can be supported by outcomes of the selection model when deciding on crossmatch tests. This two-stage formulation also allows for flexibility in the testing strategy. Although we have only chosen to impose a global bound on the number of tests in the first stage, one can also envision testing constraints per donor and recipient, differentiation between types of donor-patient pairs, and so on.

Furthermore, we experimented extensively with different implementations of the model and with different algorithms for solving it. From our experiments, it appears that Benders decomposition applied to a strengthened formulation tends to run faster than the other implementations, although simply running CPLEX's standard branch-and-cut algorithm provides competitive results. In both cases, increasing the maximum cycle length, as well as allowing chains, has a strong negative impact on the running times.

In summary, the selection model is able to yield exchange plans with a higher expected number of transplants than earlier recourse models and provides an attractive opportunity for the optimization of kidney exchange programs. However, solving this problem to optimality remains a computational challenge. Further research will be needed to develop more efficient exact or heuristic procedures for its solution.

Acknowledgments

A very preliminary version of this work was presented at the MATCH-UP 2019 conference. The final version has benefited from the comments of anonymous reviewers. In addition, discussions with participants of meetings induced by the COST-action CA15210 European Network on Collaboration for Kidney Exchange Programmes have usefully contributed to the paper. Much of this research was carried out while B. Smeulders was a postdoctoral fellow of the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) at HEC Management School, University of Liège.

Endnote

[1] All graphs are available through https://github.com/BSmeulders/RecourseinKEP.

References

Abraham DJ, Blum A, Sandholm T (2007) Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges. *Proc. 8th ACM Conf. Econom. Commerce* (ACM, New York), 295–304.

Alvelos F, Klimentova X, Rais A, Viana A (2015) A compact formulation for maximizing the expected number of transplants in kidney exchange programs. *Journal of Physics: Conference Series*, vol. 616 (IOP Publishing), 012011.

Anderson R, Ashlagi I, Gamarnik D, Rees M, Roth AE, Sönmez T, Ünver MU (2015) Kidney exchange and the alliance for paired donation: Operations research changes the way kidneys are transplanted. *Interfaces* 45(1):26–42.

Assadi S, Khanna S, Li Y (2016) The stochastic matching problem with (very) few queries. *Proc. 2016 ACM Conf. Econom. Comput.* (ACM, New York), 43–60.

Behnezhad S, Farhadi A, Hajiaghayi M, Reyhani N (2019) Stochastic matching with few queries: New algorithms and tools. *Proc. 30th Annual ACM-SIAM Sympos. Discrete Algorithms* (SIAM, Philadelphia), 2855–2874.

Biró P, Haase-Kromwijk B, Andersson T, Ásgeirsson EI, Baltesová T, Boletis I, Bolotinha C, et al (2019) Building kidney exchange programmes in Europe: An overview of exchange practice and activities. *Transplantation* 103:1514–1522.

Blum A, Gupta A, Procaccia A, Sharma A (2013) Harnessing the power of two crossmatches. *Proc. 14th ACM Conf. Electronic Commerce* (ACM, New York), 123–140.

Blum A, Dickerson JP, Haghtalab N, Procaccia AD, Sandholm T, Sharma A (2015) Ignorance is almost bliss: Near-optimal stochastic matching with few queries. *Proc. 16th ACM Conf. on Econom. Comput.* (ACM, New York), 325–342.

Dickerson JP, Procaccia AD, Sandholm T (2013) Failure-aware kidney exchange. *Proc. 14th ACM Conf. Electronic Commerce* (ACM, New York), 323–340.

Dickerson JP, Procaccia AD, Sandholm T (2019) Failure-aware kidney exchange. *Management Sci.* 65:1768–1791.

Dickerson JP, Manlove DF, Plaut B, Sandholm T, Trimble J (2016) Position-indexed formulations for kidney exchange. *Proc. 2016 ACM Conf. Econom. Comput.* (ACM, New York), 25–42.

Gentry SE, Mongomery RA, Segev DL (2011) Kidney paired donation: Fundamentals, limitations, and expansions. *Amer. J. Kidney Disease* 57(1):144–151.

Glorie K (2012) Estimating the probability of positive crossmatch after negative virtual crossmatch. Technical report, Econometric Institute, Erasmus University, Rotterdam, Netherlands.

Glorie K (2014) Clearing barter exchange markets: Kidney exchange and beyond. PhD thesis, Erasmus University, Rotterdam, Netherlands.

Homem-de-Mello T, Bayraksan G (2014) Monte Carlo sampling-based methods for stochastic optimization. *Survey Oper. Res. Management Sci.* 19:56–85.

Kleywegt A, Shapiro A, Homem-de-Mello T (2001) The sample average approximation method for stochastic discrete optimization. *SIAM J. Optim.* 12(2):479–502.

Klimentova X, Pedroso JP, Viana A (2016) Maximising expectation of the number of transplants in kidney exchange programmes. *Comput. Oper. Res.* 73:1–11.

Mak-Hau V (2017) On the kidney exchange problem: Cardinality constrained cycle and chain problems on directed graphs: A survey of integer programming approaches. *J. Combinational Optim.* 33(1):35–59.

Manlove DL, O'malley G (2015) Paired and altruistic kidney donation in the UK: Algorithms and experimentation. *J. Experiment. Algorithmics* 19:2–6.

Montgomery RA, Gentry SE, Marks WH, Warren DS, Hiller J, Houp J, Zachary AA, et al. (2006) Domino paired kidney donation: A strategy to make best use of live non-directed donation. *Lancet* 368(9533):419–421.

NHS (2017a) Annual report on living donor kidney transplantation. Technical report, NHS, London.

NHS (2017b) Living donor kidney matching run process. Technical report. Retrieved April 26, 2018, https://nhsbtdbe.blob.core.windows.net/umbraco-assets-corp/2287/ldkmr-matching-process.pdf.

Pedroso JP (2014) Maximizing expectation on vertex-disjoint cycle packing. *Proc. Internat. Conf. Comput. Sci. Appl.* (Springer, Berlin), 32–46.

Rahmaniani R, Crainic TG, Gendreau M, Rei W (2017) The Benders decomposition algorithm: A literature review. *Eur. J. Oper. Res.* 259(3):801–817.

Roth AE, Sönmez T, Ünver MU (2004) Kidney exchange. *Quart. J. Econom.* 119(2):457–488.

Roth AE, Sönmez T, Ünver MU, Delmonico FL, Saidman SL (2006) Utilizing list exchange and nondirected donation through 'chain' paired kidney donations. *Amer. J. Transplantation* 6(11): 2694–2705.

Saidman SL, Roth AL, Sönmez T, Ünver MU, Delmonico FL (2006) Increasing the opportunity of live kidney donation by matching for two-and three-way exchanges. *Transplantation* 81(5):773–782.